Mini Project
On
YouTube Video Analysis Web Application

Jawaharlal Nehru Technological University Anantapur, Ananthapuramu

In partial fulfillment of the requirements
for the award of the degree of

**BACHELOR OF TECHNOLOGY IN
INFORMATION TECHNOLOGY**

Submitted by

21125A1212	***S. Mahesh Babu***
21125A1213	***S. Karthika***

Department of Information Technology

SREE VIDYANIKETHAN ENGINEERING COLLEGE
(AUTONOMOUS)

(Affiliated to JNTUA, Ananthapuramu, Approved by AICTE, Accredited by NBA& NAAC)
Sree Sainath Nagar,Tirupati–517102,A.P.,INDIA

2022-2023

BOTTLE

It is a fast, simple and lightweight WSGI micro web-framework for Python.

What is Bottle?

Bottle is a lightweight and straightforward micro web framework for Python, designed for rapid development of small web applications. Its minimalistic design allows developers to quickly create web services, APIs, or simple web pages using a compact and easily deployable framework. Bottle provides routing, templating, and a basic HTTP server, making it well-suited for small-scale projects where simplicity and minimal overhead are essential. Its single-file nature and Pythonic approach make it an attractive choice for those seeking a fast and uncomplicated solution for web development tasks.
You can check details of it in the below link:https://bottlepy.org/docs/dev/

INSTALLATION:

Installation of Bottle on Windows

1. Python:

Ensure you have Python installed. If not, download and install it from the official Python website: https://www.python.org/downloads/

2. Bottle:

Install the Bottle web framework using pip, the Python package manager:

```
>>> pip install bottle
```

3. yt-dlp:

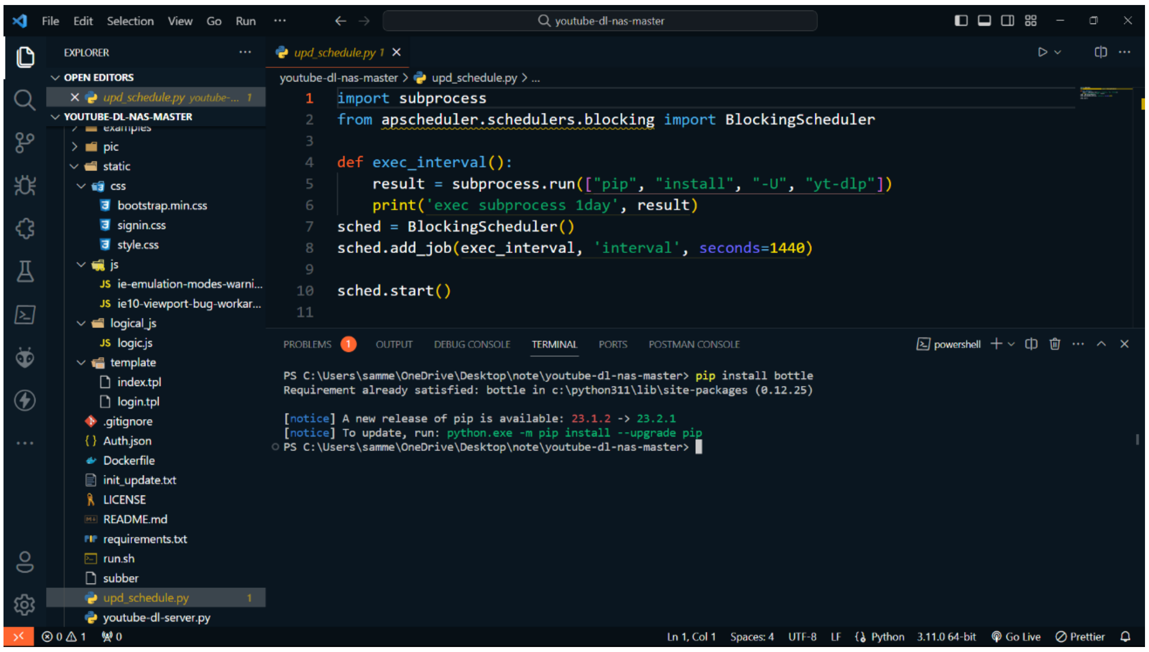
```
>>> pip install yt-dlp

>>> pip install textblob

>>>  python -m textblob.download_corpora

>>> pip install pytrends

>>>  pip install matplotlib
```



```python
import subprocess
from apscheduler.schedulers.blocking import BlockingScheduler

def exec_interval():
    result = subprocess.run(["pip", "install", "-U", "yt-dlp"])
    print('exec subprocess 1day', result)
sched = BlockingScheduler()
sched.add_job(exec_interval, 'interval', seconds=1440)

sched.start()
```

```
PS C:\Users\samme\OneDrive\Desktop\note\youtube-dl-nas-master> pip install bottle
Requirement already satisfied: bottle in c:\python311\lib\site-packages (0.12.25)

[notice] A new release of pip is available: 23.1.2 -> 23.2.1
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\samme\OneDrive\Desktop\note\youtube-dl-nas-master>
```

Fig 1:Installing bottle in VS code.

4. Template Files:

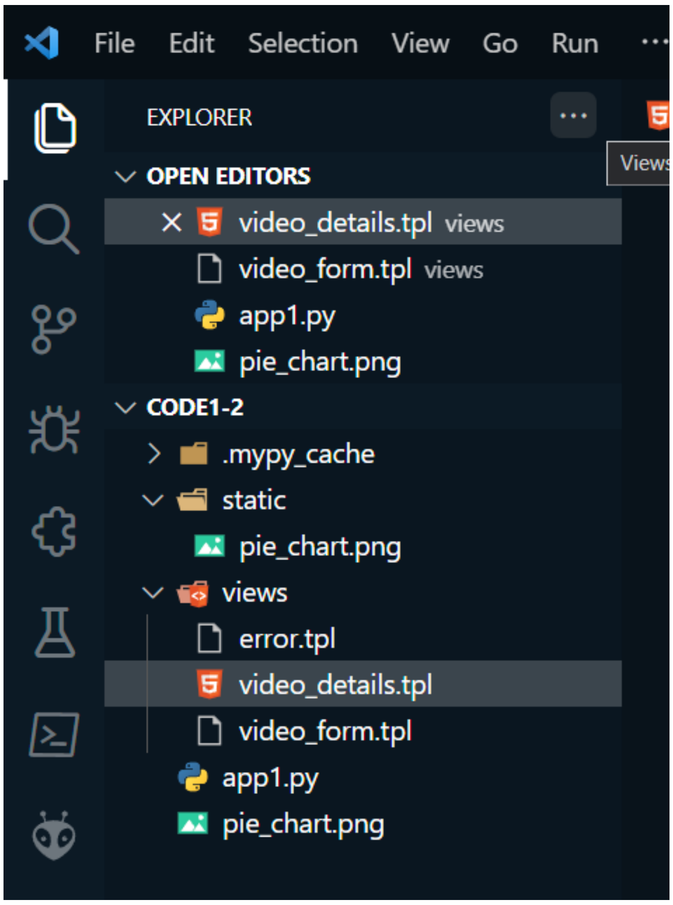
Create a directory named views in your project folder.
Save the video_form.tpl and video_details.tpl templates in the views directory.

5. Static Files:

Create a directory named static in your project folder.
Save the pie_chart.png image in the static directory.

6. Application Code:

Create an app.py file in your project directory.
Add the application code to this file.



```
OPEN EDITORS
    × video_details.tpl views
      video_form.tpl views
      app1.py
      pie_chart.png
CODE1-2
  > .mypy_cache
  ∨ static
      pie_chart.png
  ∨ views
      error.tpl
      video_details.tpl
      video_form.tpl
    app1.py
    pie_chart.png
```

Fig 2 : File Structure in this project.

USECASE:

Title : YouTube Video Analysis Web Application

The case study examines the development and implementation of a Bottle Python web application for YouTube video analysis, designed to empower digital marketing professionals to efficiently extract and analyse key video metrics. This study details the project's inception, scope, development process, features, and its impact on XYZ Marketing Agency, shedding light on the value it brings to the digital marketing industry.

Streamline Video Analysis: To simplify and accelerate the process of extracting and understanding YouTube video metrics for digital marketing professionals.
Provide Actionable Insights: To offer data-driven insights that help users optimize content and engage their audience more effectively.

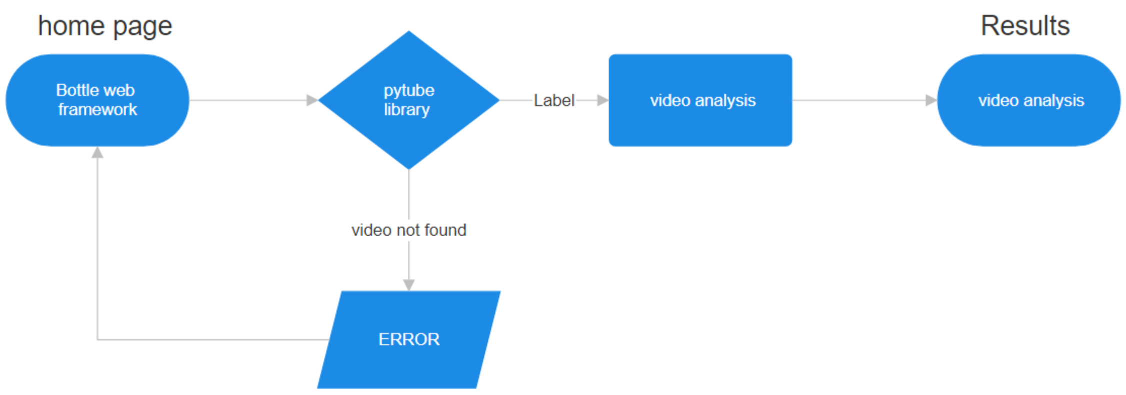


Fig 3 : Workflow of Bottle

Advantage of Bottle :

- Simplicity
- Simplicity
- Lightweight
- Fast:
- Embeddable
- No External Templates or Databases
- Good for Learning

CODE:

app.py

```
from bottle import route, run, request, template
from yt_dlp import YoutubeDL
from datetime import datetime

# Bottle route to handle the form input and display results
@route("/", method="GET")
def video_analysis():
    # Check if the form has been submitted
    if request.GET.get("submit"):
        video_url = request.GET.get("video_url")

        try:
            # Function to fetch video details using yt-dlp
            def get_video_details(video_url):
                ydl_opts = {}
                with YoutubeDL(ydl_opts) as ydl:
                    info = ydl.extract_info(video_url, download=False)
```

```
# Explicitly convert the numeric values to integers
views = int(info.get("view_count", 0))
likes = int(info.get("like_count", 0))
dislikes = int(info.get("dislike_count", 0))
title = info.get("title", "N/A")
author = info.get("uploader", "N/A")
description = info.get("description", "N/A")
keywords = info.get("keywords", "N/A")
publish_date = datetime.strptime(
    info.get("upload_date", "N/A"), "%Y%m%d"
)
length = int(info.get("duration", 0))
thumbnail_url = info.get("thumbnail", "N/A")

# Calculate like-to-dislike ratio
like_dislike_ratio = (likes / (likes + dislikes)) * 100

# Calculate engagement rate
engagement_rate = ((likes + dislikes) / views) * 100

# Calculate days since publish
days_since_publish = (datetime.now() - publish_date).days

# Determine rating category
if likes >= dislikes * 10:
    rating_category = "Excellent"
elif likes >= dislikes * 5:
    rating_category = "Good"
elif likes >= dislikes:
    rating_category = "Fair"
else:
    rating_category = "Poor"

return template(
    "video_details",
    title=title,
    author=author,
    description=description,
    views=views,
    likes=likes,
    dislikes=dislikes,
    publish_date=publish_date,
    keywords=keywords,
    length=length,
    thumbnail_url=thumbnail_url,
    like_dislike_ratio=like_dislike_ratio,
    engagement_rate=engagement_rate,
    days_since_publish=days_since_publish,
    rating_category=rating_category,
)
```

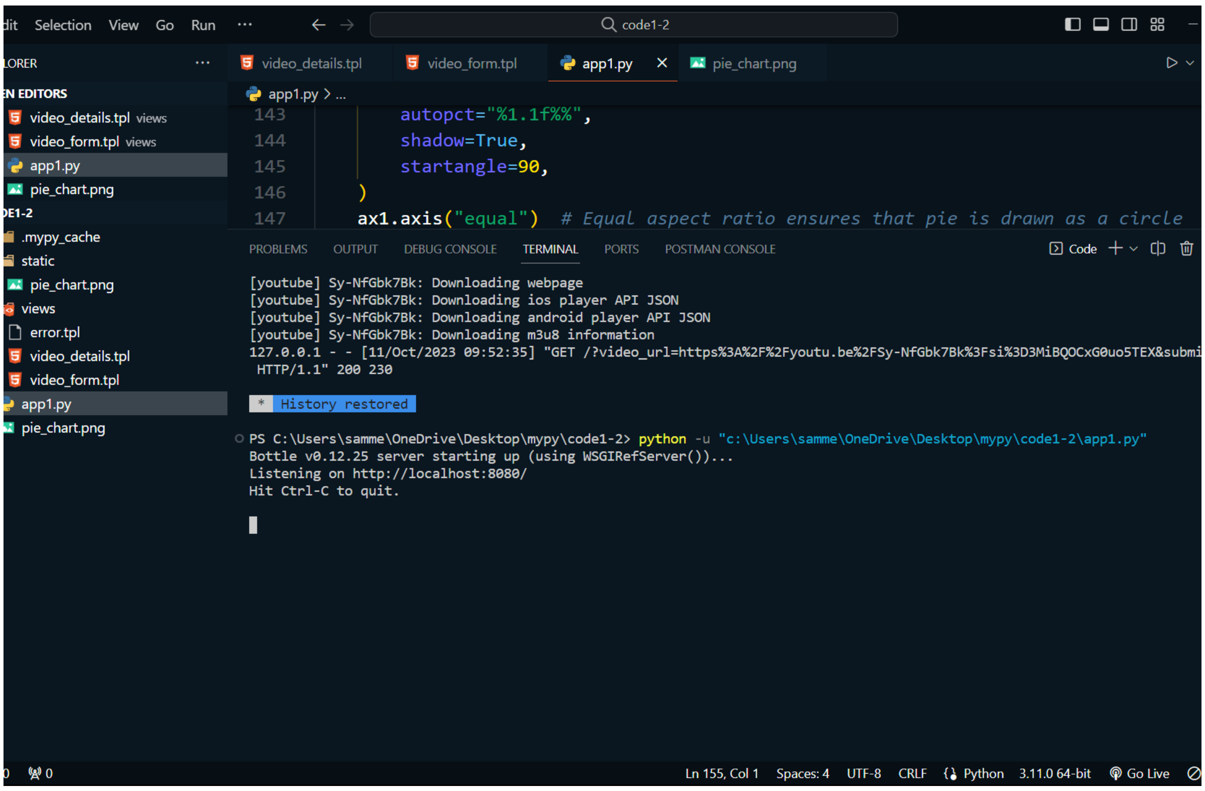
```
            return "Error: Video details not found."

        return get_video_details(video_url)

    except Exception as e:
        error_message = f"Error: {str(e)}"
        return template("error", error_message=error_message)

  # If the form hasn't been submitted, render the input form
  return template("video_form")


if __name__ == "__main__":
  run(host="localhost", port=8080)
```



```
            autopct="%1.1f%%",
            shadow=True,
            startangle=90,
        )
        ax1.axis("equal")  # Equal aspect ratio ensures that pie is drawn as a circle
```

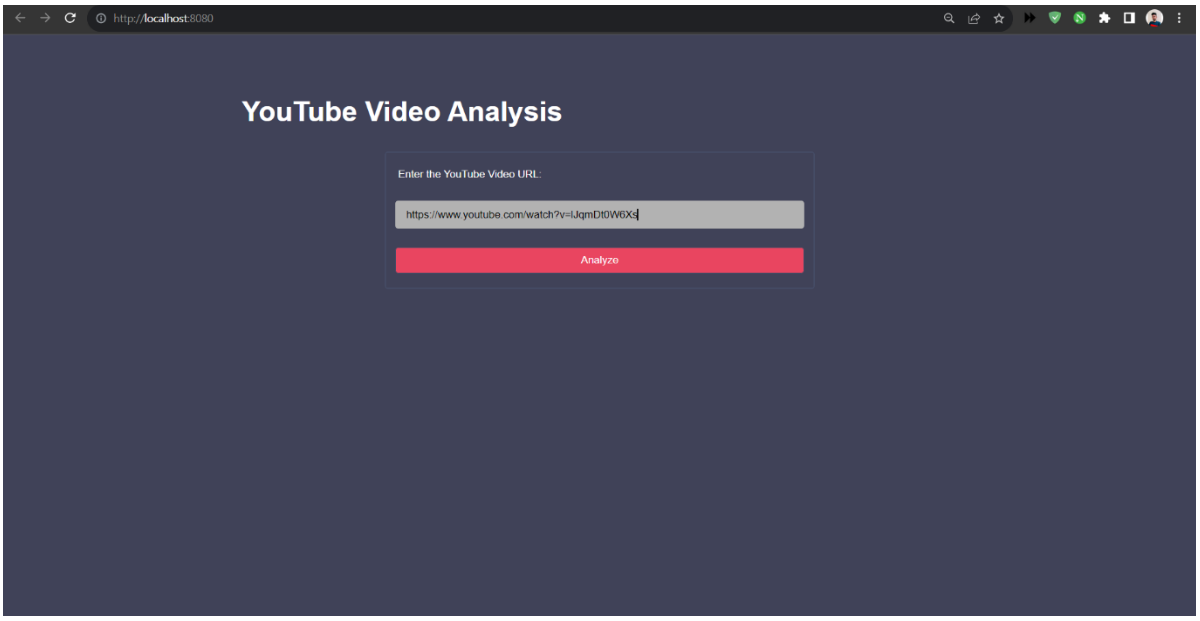
```
[youtube] Sy-NfGbk7Bk: Downloading webpage
[youtube] Sy-NfGbk7Bk: Downloading ios player API JSON
[youtube] Sy-NfGbk7Bk: Downloading android player API JSON
[youtube] Sy-NfGbk7Bk: Downloading m3u8 information
127.0.0.1 - - [11/Oct/2023 09:52:35] "GET /?video_url=https%3A%2F%2Fyoutu.be%2FSy-NfGbk7Bk%3Fsi%3D3MiBQOCxG0uo5TEX&submi
 HTTP/1.1" 200 230

 *  History restored

PS C:\Users\samme\OneDrive\Desktop\mypy\code1-2> python -u "c:\Users\samme\OneDrive\Desktop\mypy\code1-2\app1.py"
Bottle v0.12.25 server starting up (using WSGIRefServer())...
Listening on http://localhost:8080/
Hit Ctrl-C to quit.
```

Fig 4 :Running of app in the web browser

Output:

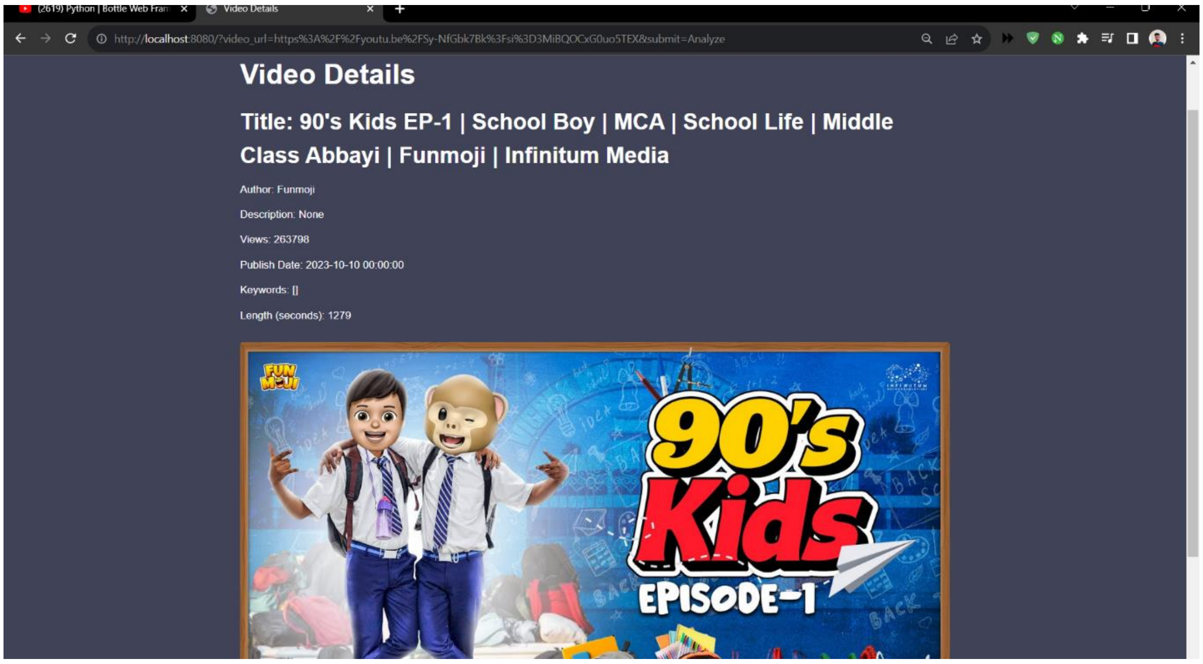


YouTube Video Analysis

Enter the YouTube Video URL:

https://www.youtube.com/watch?v=lJqmDt0W6Xs

Fig 5 : Home page of project.


(2619) Python | Bottle Web Fram
Video Details
http://localhost:8080/?video_url=https%3A%2F%2Fyoutu.be%2FSy-NfGbk7Bk%3Fsi%3D3MiBQOCxG0uo5TEX&submit=Analyze
FUN MOJI
90's Kids
EPISODE-1

Video Details

Title: 90's Kids EP-1 | School Boy | MCA | School Life | Middle Class Abbayi | Funmoji | Infinitum Media

Author: Funmoji

Description: None

Views: 263798

Publish Date: 2023-10-10 00:00:00

Keywords: []

Length (seconds): 1279

Fig 6 : Analysis results.